### \*\*思考题答案\*\*

#### 一、发送速率与接收速率不匹配

如果A的发送速率比B的接收速率快，会出现\*\*拥塞\*\*问题，B来不及处理所有接收到的数据帧，导致部分数据被丢弃或延迟处理。

\*\*解决方法：\*\*

1. \*\*流量控制 (Flow Control)：\*\*

- 使用\*\*滑动窗口协议\*\*限制A发送的数据量，让A发送一定数量的数据帧后等待B的确认（ACK）。

2. \*\*停止等待协议 (Stop-and-Wait)：\*\*

- 每发送一个数据帧，A等待B确认后再发送下一个。

·缺陷：效率低，解决方法：多发几次

3. \*\*拥塞控制 (Congestion Control)：\*\*

- 利用如\*\*TCP\*\*中的慢启动和拥塞避免机制，动态调整发送速率。

---

#### 二、如何实现可靠的数据传输服务

1. \*\*如何发现错误？\*\*

- \*\*校验机制 (Checksum)：\*\* 每个数据帧附加校验和，B接收到数据后重新计算校验和并比对，检查是否一致。

- \*\*循环冗余校验 (CRC)：\*\* 提高检测效率，更适用于数据链路层。

2. \*\*如何纠正错误？\*\*

- \*\*自动重传请求 (ARQ)：\*\* B检测到错误后请求A重传错误帧。

- \*\*前向纠错 (FEC)：\*\* 数据帧中添加冗余信息，B可自行修正部分错误。

3. \*\*如何发现丢包？\*\*

- \*\*超时重传机制：\*\* A在发送数据帧后启动计时器，如果未收到B的确认帧（ACK），则判断数据可能丢失并重新发送。

4. \*\*如何判断重复的报文？\*\*

- 每个数据帧附带\*\*序列号\*\*，B通过比对序列号判断是否为重复报文。例如，若B已收到序列号为5的帧，则新收到的5号帧会被识别为重复并丢弃。

010101.。。这样

---

#### 三、适合A、B之间传输的数据帧结构设计

| 字段 | 长度 | 描述 |

|---------------|--------|-------------------------------|

| 帧头 | 固定长度 | 标志帧的起始 |

| 序列号 | N位 | 标识数据帧的顺序 |

| 数据部分 | 可变长度 | 传输的实际数据 |

| 校验和 (CRC) | 固定长度 | 用于错误检测 |

| 帧尾 | 固定长度 | 标志帧的结束 |

---

#### 四、完整的工作协议设计

\*\*发送方 A 的协议：\*\*

1. 准备要发送的数据帧并计算校验和。

2. 发送数据帧后启动计时器，等待确认帧（ACK）。

3. 如果收到ACK，则发送下一个数据帧；如果超时未收到ACK，则重传当前数据帧。

4. 当所有数据发送完成，发送结束信号。

\*\*接收方 B 的协议：\*\*

1. 接收数据帧并计算校验和。

2. 若校验通过，发送ACK并将数据传递给上层；若校验失败，丢弃数据帧并请求重传。

3. 维护一个序列号表，判断是否为重复帧，若为重复则丢弃。

4. 若接收到结束信号，则关闭连接。

---

这个方案通过流量控制和错误检测机制保证了A、B之间的可靠通信，同时设计了合理的数据帧结构和协议流程，确保数据传输的准确性和完整性。

根据图片中的发送方算法，接收方的算法可以这样设计：

### \*\*接收方算法：\*\*

1. \*\*等待数据帧到达：\*\*

- 持续监听来自发送方的消息。

2. \*\*检测收到的数据帧：\*\*

- 检查数据帧的完整性（如\*\*校验和\*\*或\*\*CRC\*\*）。

- 如果数据帧有误，直接丢弃，并\*\*不发送确认帧\*\*（或发送否定确认帧NAK，若协议支持）。

- 如果数据帧无误，执行下一步。

3. \*\*检查序列号或标识符：\*\*

- 检查数据帧的\*\*序列号\*\*是否与预期一致。

- 如果序列号正确（即是预期的数据帧），接收方将数据传递给上层应用，并进入下一步。

- 如果序列号错误（例如重复数据），则丢弃该帧并重新等待新的数据帧。

4. \*\*发送确认帧 (ACK)：\*\*

- 向发送方发送\*\*ACK\*\*（确认帧），表明该数据帧已正确接收。

5. \*\*继续接收下一数据帧：\*\*

- 重复上述过程，等待并处理下一帧。

---

### \*\*流程示例：\*\*

1. \*\*收到正确数据帧：\*\*

- 数据帧校验成功，序列号正确，发送ACK，传递数据给应用层。

2. \*\*收到错误数据帧：\*\*

- 数据帧校验失败，丢弃数据，不发送ACK，等待重传。

3. \*\*收到重复数据帧：\*\*

- 序列号与已确认的帧相同，丢弃重复帧，发送ACK确认已接收。

通过这种接收方算法，可以确保：

- \*\*数据完整性\*\*（通过校验和/CRC）。

- \*\*数据顺序一致\*\*（通过序列号）。

- \*\*数据可靠传输\*\*（通过ACK机制请求重传或确认）。